EXPRESS MAIL LABEL NO.: EL563155585US

CLAIMS

1. A method for authenticating an external module comprising the steps of:

creating data K using two different schemes, at least one scheme being based on the integrity of the module to be verified;

creating an authentication token for said module which produces K in both schemes;

using K as created by one scheme to disrupt said module; and

using K as created by the other scheme to restore the module.

2. The method as defined in claim 1, wherein one or more of the schemes is based on RSA encryption.

3. The method as defined in claim 1, wherein one or more of the schemes is based on digital signets.

4. The method as defined in claim 1, further comprising the step of:

embedding a public component of one or more of the schemes in a verification code of the application.

5. The method as defined in claim 1, further comprising the step of:

conveying private components of all schemes to a module authentication authority.

6. The method as defined in claim 5, wherein the creating an authentication token step is automated by providing developers of the external module access to a Web site which allows them to submit the hash of their module and to retrieve the authentication token.

7. The method as defined in claim 1, wherein the authentication token is embedded in the external module as long as the external module itself remains functional.

8. The method as defined in claim 7, where the embedding is realized by adding an additional data section to a DLL in Portable Executable (PE) format.

9. The method as defined in claim 1, wherein the authentication token is external to the external module.

10. The method as defined in claim 1, wherein said scheme depending on code integrity is independent of a location of the external module in memory.

11. The method as defined in claim 10, wherein location independence is achieved by locating and reading the external module's image on a disk.

12. The method as defined in claim 10, wherein location independence is achieved by using a canonical hash.

13. A method for secure authentication of external modules on an entity comprising the steps of:

loading an external module into memory; and

beginning a STOMPing process by decrypting a number of pseudo-random bytes that are part of an authentication token using a public security code of a public and private component pair security code.

14. The method as defined in claim 13, wherein the public and private components of the security code comprise security codes selected from a group of security codes of: a signet pair and an RSA pair.

15. The method as defined in claim 13, further comprising the step of:

XORing the decrypted pseudo-random bytes with the external module making the external module unusable.

16. The method as defined in claim 15, further comprising the step of:

performing a signet extrication process to generate extrication data by using the hash of the external module to begin an UNSTOMPing process.

17. The method as defined in claim 16, further comprising the step of:

using the extrication data to generate another stream of pseudo-random bytes.

18. The method as defined in claim 17, further comprising the step of:

XORing the another stream of pseudo-random bytes with the unusable external module thereby making the unusable external module usable in the event there has been no illicit patching of the external module and maintaining the unusable external module unusable in the event that the external module has been illicitly patched such that an application or program that is accessing the module fails to operate.

19. The method as defined in claim 18, further comprising the step of:

re-authenticating the external module by periodically performing the STOMP and UNSTOMPing process multiple times while interacting with the external module.

20. The method as defined in claim 18, further comprising the step of: performing run time checks to make sure that function calls to the external module are not intercepted by an attacker.

EXPRESS MAIL LABEL NO.: EL563155585US

21. A computer readable medium comprising instructions for secure authentication of external modules on an entity comprising the instructions of:

loading an external module into memory; and

beginning a STOMPing process by decrypting a number of pseudo-random bytes that are part of an authentication token using a public security code of a public and private component pair security code.

22. The computer readable medium as defined in claim 21 wherein the public and private components of the security code comprise security codes selected from a group of security codes of: a signet pair and an RSA pair.

23. The computer readable medium as defined in claim 21 further comprising the instruction of:

XORing the decrypted pseudo-random bytes with the external module making the external module unusable.

24. The computer readable medium as defined in claim 23 further comprising the instruction of:

performing a signet extrication process to generate extrication data by using the hash of the external module to begin an UNSTOMPing process.

25. The computer readable medium as defined in claim 24 further comprising the instruction of:

using the extrication data to generate another stream of pseudo-random bytes.

26. The computer readable medium as defined in claim 25 further comprising the instruction of:

XORing the another stream of pseudo-random bytes with the unusable external module thereby making the unusable external module usable in the event there has been no illicit patching of the external module and maintaining the unusable external module unusable in the event that the external module has been illicitly patched such that an application or program that is accessing the module fails to operate.

27. A system for secure authentication of external modules on an entity comprising the units of:

a loader unit for loading an external module into memory; and beginning a STOMPing process by

a decryption unit for decrypting a number of pseudo-random bytes that are part of an authentication token using a public security code of a public and private component pair security code.